MPLS WG
Internet Draft
Document: draft-ietf-mpls-ldp-ft-00.txt
Expiration Date: March 2001

Adrian Farrel
Paul Brittain
Data Connection Ltd

Philip Matthews
Nortel

Eric Gray
Zaffire

October 2000

Fault Tolerance for LDP and CR-LDP

Status of this Memo

NOTE: The new TLV type numbers, bit values for flags specified in this draft, and new LDP status code values are preliminary suggested values and have yet to be approved by IANA or the MPLS WG. See the section "IANA Considerations" for further details.

Abstract

MPLS systems will be used in core networks where system downtime must be kept to an absolute minimum. Many MPLS LSRs may, therefore, exploit Fault Tolerant (FT) hardware or software to provide high availability of the core networks.

The details of how FT is achieved for the various components of an FT LSR, including LDP, CR-LDP, the switching hardware and TCP, are implementation specific. This document identifies issues in the CR-LDP specification [2] and the LDP specification [4] that make it difficult to implement an FT LSR using the current LDP and CR-LDP

protocols, and proposes enhancements to the LDP specification to ease such FT LSR implementations.

The extensions described here are equally applicable to CR-LDP.

Contents

1. Conventions and Terminology used in this document

Definitions of key words and terms applicable to LDP and CR-LDP are inherited from [2] and [4].

The term "FT label" is introduced in this document to indicated a label for which fault tolerant operation is used. A "non-FT label" is not fault tolerant and is handled as specified in [2] and [4].

The extensions to LDP specified in this document are collectively referred to as the "LDP FT enhancements".

In the examples quoted, the following notation is used.

Ln : An LSP. For example L1.
Pn : An LDP peer. For example P1.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [3].

2. Introduction

High Availability (HA) is typically claimed by equipment vendors when their hardware achieves availability levels of at least 99.999% (five 9s). To implement this, the equipment must be capable of recovering from local hardware and software failures through a process known as fault tolerance (FT).

The usual approach to FT involves provisioning backup copies of hardware and software. When a primary copy fails, processing is switched to the backup copy. This process, called failover, should result in minimal disruption to the Data Plane.

In an FT system, backup resources are sometimes provisioned on a one-to-one basis (1:1), sometimes as many-to-one (1:n), and occasionally as many-to-many (m:n). Whatever backup provisioning is made, the system must switch to the backup automatically on failure of the primary, and the software and hardware state in the backup must be set to replicate the state in the primary at the point of failure.

2.1 Fault Tolerance for MPLS

MPLS will be used in core networks where system downtime must be kept to an absolute minimum. Many MPLS LSRs may, therefore, exploit FT

hardware or software to provide high availability of core networks.

In order to provide HA, an MPLS system needs to be able to survive a variety of faults with minimal disruption to the Data Plane, including the following fault types:

- failure/hot-swap of a physical connection between LSRs
- failure/hot-swap of the switching fabric in the LSR
- failure of the TCP or LDP stack in an LSR
- software upgrade to the TCP or LDP stacks.

The first two examples of faults listed above are confined to the Data Plane. Such faults can be handled by providing redundancy in the Data Plane which is transparent to LDP operating in the Control Plane. The last two example types of fault require action in the Control Plane to recover from the fault without disrupting traffic in the Data Plane. This is possible because many recent router architectures separate the Control and Data Planes such that forwarding can continue unaffected by recovery action in the Control Plane.

2.2 Issues with LDP and CR-LDP

LDP and CR-LDP use TCP to provide reliable connections between LSRs over which to exchange protocol messages to distribute labels and to set up LSPs. A pair of LSRs that have such a connection are referred to as LDP peers.

TCP enables LDP and CR-LDP to assume reliable transfer of protocol messages. This means that some of the messages do not need to be acknowledged (for example, Label Release).

LDP and CR-LDP are defined such that if the TCP connection fails, the LSR should immediately tear down the LSPs associated with the session between the LDP peers, and release any labels and resources assigned to those LSPs.

It is notoriously hard to provide a Fault Tolerant implementation of TCP. To do so might involve making copies of all data sent and received. This is an issue familiar to implementers of other TCP applications such as BGP.

During failover affecting the TCP or LDP stacks, therefore, the TCP connection may be lost. Recovery from this position is made worse by the fact that LDP or CR-LDP control messages may have been lost during the connection failure. Since these messages are unconfirmed, it is possible that LSP or label state information will be lost.

This draft describes a solution which involves
- negotiation between LDP peers of the intent to support extensions to LDP that facilitate recovery from failover without loss of LSPs
- selection of FT survival on a per LSP/label basis
- acknowledgement of LDP messages to ensure that a full handshake is performed on those messages
- re-issuing lost messages after failover to ensure that LSP/label state is correctly recovered after reconnection of the LDP session.

Other objectives of this draft are to
- offer back-compatibility with LSRs that do not implement these proposals
- preserve existing protocol rules described in [2] and [4] for handling unexpected duplicate messages and for processing unexpected messages referring to unknown LSPs/labels
- integrate with the LSP modification function described in [5]
- avoid full state refresh solutions (such as those present in RSVP: see [6], [7] and [8]) whether they be full-time, or limited to post-failover recovery.

Note that this draft concentrates on the preservation of label state for labels exchanged between a pair of adjacent LSRs when the TCP connection between those LSRs is lost. The is a requirement for Fault Tolerant operation of LSPs, but a full implementation of end-to-end protection for LSPs requires that this is combined with other techniques that are outside the scope of this draft.

In particular, this draft does not attempt to describe how to modify the routing of an LSP or the resources allocated to a label or LSP, which is covered by [5]. This draft also does not address how to provide automatic layer 2/3 protection switching for a label or LSP, which is a separate area for study.

3. Overview of LDP FT Enhancements

The LDP FT enhancements consist of the following main elements, which are described in more detail in the sections that follow.

- The presence of an FT Session TLV on the LDP Initialization message indicates that an LSR supports the LDP FT enhancements on this session.

- An FT Reconnect Flag in the FT Session TLV indicates whether an LSR has preserved FT label state across a failure of the TCP connection.

- An FT Reconnection Timeout, exchanged on the LDP Initialization message, that indicates the maximum time peer LSRs will preserve

FT label state after a failure of the TCP connection.

- An FT Protection TLV used to identify operations that affect LDP labels. All LDP messages carrying the FT Protection TLV need to be secured (e.g. to NVRAM) and ACKed to the sending LDP peer in order that the state for FT labels can be correctly recovered after LDP session reconnection.

3.1 Establishing an FT LDP Session

In order that the extensions to LDP [4] and CR-LDP [2] described in this draft can be used successfully on an LDP session between a pair of LDP peers, they MUST negotiate that the LDP FT enhancements are to be used on the LDP session.

This is done on the LDP Initialization message exchange using a new FT Session TLV. Presence of this TLV indicates that the peer wants to support the LDP FT enhancements on this LDP session.

The LDP FT enhancements MUST be supported on an LDP session if both LDP peers include an FT Session TLV on the LDP Initialization message.

If either LDP Peer does not include the FT Session TLV on the LDP Initialization message, the LDP FT enhancements MUST NOT be used on the LDP session.

An LSR MAY present different FT/non-FT behavior on different TCP connections, even if those connections are successive instantiations of the LDP session between the same LDP peers.

3.1.1 Interoperation with Non-FT LSRs

The FT Session TLV on the LDP Initialization message carries the U-bit. If an LSR does not support the LDP FT enhancements, it will ignore this TLV. Since such partners also do not include the FT Session TLV, all LDP sessions to such LSRs will not use the LDP FT enhancements.

The rest of this draft assumes that the LDP sessions under discussion are between LSRs that do support the LDP FT enhancements, except where explicitly stated otherwise.

3.2 TCP Connection Failure

If the LDP FT enhancements are not in use on an LDP session, the action of the LDP peers on failure of the TCP connection is as specified in [2] and [4].

All state information and resources associated with non-FT labels MUST be released on the failure of the TCP connection, including deprogramming the non-FT label from the switching hardware. This is equivalent to the behavior specified in [4].

If the LDP FT enhancements are in use on an LDP session, both LDP peers SHOULD preserve state information and resources associated with FT labels exchanged on the LDP session. Both LDP peers SHOULD use a timer to release the preserved state information and resources associated with FT-labels if the TCP connection is not restored within a reasonable period. The behavior when this timer expires is equivalent to the LDP session failure behavior described in [4].

The FT Reconnection Timeout each LDP peer intends to apply to the LDP session is carried in the FT Session TLV on the LDP Initialization messages. It is RECOMMENDED that both LDP peers use the lower timeout value from the LDP Initialization exchange when setting their reconnection timer after a TCP connection failure.

3.3 Data Forwarding During TCP Connection Failure

An LSR that implements the LDP FT enhancements SHOULD preserve the programming of the switching hardware across a failover. This ensures that data forwarding is unaffected by the state of the TCP connection between LSRs.

It is an integral part of FT failover processing in some hardware configurations that some data packets might be lost. If data loss is not acceptable to the applications using the MPLS network, the LDP FT enhancements described in this draft SHOULD NOT be used.

3.4 FT LDP Session Reconnection

When a new TCP connection is established, the LDP peers MUST exchange LDP Initialization messages. When a new TCP connection is established after failure, the LDP peers MUST re-exchange LDP Initialization messages.

If an LDP peer includes the FT Session TLV in the LDP Initialization message for the new instantiation of the LDP session, it MUST also set the FT Reconnect Flag according to whether it has been able to preserve label state. The FT Reconnect Flag is carried in the FT Session TLV.

If an LDP peer has preserved all state information for previous instantiations of the LDP session, then it SHOULD set the FT Reconnect Flag to 1 in the FT Session TLV. Otherwise, it MUST set the

FT Reconnect Flag to 0.

If either LDP peer sets the FT Reconnect Flag to 0, or omits the FT Session TLV, both LDP peers MUST release any state information and resources associated with the previous instantiation of the LDP session between the same LDP peers, including FT label state and Addresses. This ensures that network resources are not permanently lost by one LSR if its LDP peer is forced to undergo a cold start.

If both LDP peers set the FT Reconnect Flag to 1, both LDP peers MUST use the FT label operation procedures indicated in this draft to complete any label operations on FT labels that were interrupted by the LDP session failure.

3.5 Operations on FT Labels

Label operations on FT labels are made Fault Tolerant by providing acknowledgement of all LDP messages that affect FT labels. Acknowledgements are achieved by means of sequence numbers on these LDP messages.

The message exchanges used to achieve acknowledgement of label operations and the procedures used to complete interrupted label operations are detailed in the section "FT Operations".

Using these acknowledgements and procedures, it is not necessary for LDP peers to perform a complete re-synchronization of state for all FT labels, either on re-connection of the LDP session between the LDP peers or on a timed basis.

4. FT Operations

Once an FT LDP session has been established, using the procedures described in the section "Establishing an FT LDP Session", both LDP peers MUST apply the procedures described in this section for FT LDP message exchanges.

If the LDP session has been negotiated to not use the LDP FT enhancements, these procedures MUST NOT be used.

4.1 FT LDP Messages

4.1.1 FT Label Messages

A label is identified as being an FT label if the initial Label Request or Label Mapping message relating to that label carries the FT Protection TLV.

If a label is an FT label, all LDP messages affecting that label MUST carry the FT Protection TLV in order that the state of the label can be recovered after a failure of the LDP session.

4.1.1.1 Scope of FT Labels

The scope of the FT/non-FT status of a label is limited to the LDP message exchanges between a pair of LDP peers.

In Ordered Control, when the message is forwarded downstream or upstream, the TLV may be present or absent according to the requirements of the LSR sending the message.

4.1.2 FT Address Messages

If an LDP session uses the LDP FT enhancements, both LDP peers MUST secure Address and Address Withdraw messages using FT Operation ACKs, as described below. This avoids any ambiguity over whether an Address is still valid after the LDP session is reconnected.

If an LSR determines that an Address message that it sent on a previous instantiation of a recovered LDP session is no longer valid, it MUST explicitly issue an Address Withdraw for that address when the session is reconnected.

If the FT Reconnect Flag is not set by both LDP peers on reconnection of an LDP session (i.e. state has not been preserved), both LDP peers MUST consider all Addresses to have been withdrawn. The LDP peers SHOULD issue new Address messages for all their valid addresses as specified in [4].

4.2 FT Operation ACKs

Handshaking of FT LDP messages is achieved by use of ACKs. Correlation between the original message and the ACK is by means of the FT Sequence Number contained in the FT Protection TLV, and passed back in the FT ACK TLV. The FT ACK TLV may be carried on any LDP message that is sent on the TCP connection between LDP peers.

An LDP peer maintains a separate FT sequence number for each LDP session it participates in. The FT Sequence number is incremented by one for each FT LDP message (i.e. containing the FT Protection TLV) issued by this LSR on the FT LDP session with which the FT sequence number is associated.

When an LDP Peer receives a message containing the FT Protection TLV, it MUST take steps to secure this message (or the state information derived from processing the message). Once the message is secured, it MUST be ACKed. However, there is no requirement on the LSR to send this ACK immediately.

ACKs may be accumulated to reduce the message flow between LDP peers. For example, if an LSR received FT LDP messages with sequence numbers 1, 2, 3, 4, it could send a single ACK with sequence number 4 to ACK receipt and securing of all these messages.

ACKs MUST NOT be sent out of sequence, as this is incompatible with the use of accumulated ACKs .

4.3 Preservation of FT State

If the LDP FT enhancements are in use on an LDP session, each LDP peer SHOULD NOT release the state information and resources associated with FT labels exchanged on that LDP session when the TCP connection fails. This is contrary to [2] and [4], but allows label operations on FT labels to be completed after re-connection of the TCP connection.

Both LDP peers on an LDP session that is using the LDP FT enhancements SHOULD preserve the state information and resources they hold for that LDP session as described below.

- An upstream LDP peer SHOULD release the resources (in particular bandwidth) associated with an FT label when it initiates a Label Release or Label Abort message for the label. The upstream LDP peer MUST preserve state information for the label, even if it releases the resources associated with the label, as it may need to reissue the label operation if the TCP connection is interrupted.

- An upstream LDP peer MUST release the state information and resources associated with an FT label when it receives an acknowledgement to a Label Release or Label Abort message that it sent for the label, or when it sends a Label Release message in response to a Label Withdraw message received from the downstream LDP peer.

- A downstream LDP peer SHOULD NOT release the resources associated with an FT label when it sends a Label Withdraw message for the label as it has not yet received confirmation that the upstream LDP peer has ceased to send data using the label. The downstream LDP peer MUST NOT release the state information it holds for the label as it may yet have to reissue the label operation if the TCP connection is interrupted.

- A downstream LDP peer MUST release the resources and state information associated with an FT label when it receives an acknowledgement to a Label Withdraw message for the label.

- When the FT Reconnection Timeout expires, an LSR SHOULD release all state information and resources from previous instantiations of the (permanently) failed LDP session.

- When an LSR receives a Status TLV with the E-bit set in the status code, which causes it to close the TCP connection, the LSR MUST release all state information and resources associated with the session. This behavior is mandated because it is impossible for the LSR to predict the precise state and future behavior of the partner LSR that set the E-bit without knowledge of the implementation of that partner LSR.

  Note that the "Temporary Shutdown" status code does not have the E-bit set, and MAY be used during maintenance or upgrade operations to indicate that the LSR intends to preserve state across a closure and re-establishment of the TCP session.

- If an LSR determines that it must release state for any single FT label during a failure of the TCP connection on which that label was exchanged, it MUST release all state for all labels on the LDP session.

The release of state information and resources associated with non-FT labels is as described in [2] and [4].

Note that a Label Release and the acknowledgemet to a Label Withdraw may be received by a downstream LSR in any order. The downstream LSR MAY release its resources on receipt of the first message and MUST release its resources on receipt of the second message.

4.4 FT Procedure After TCP Failure

When an LSR discovers or is notified of a TCP connection failure it SHOULD start an FT Reconnection Timer to allow a period for re-connection of the TCP connection between the LDP peers.

Once the TCP connection between LDP peers has failed, the active LSR SHOULD attempt to re-establish the TCP connection. The mechanisms, timers and retry counts to re-establish the TCP connection are an implementation choice. It is RECOMMENDED that any attempt to re-establish the connection take account of the failover processing necessary on the peer LSR, the nature of the network between the LDP peers, and the FT Reconnection Timeout chosen on the previous instantiation of the TCP connection (if any).

If the TCP connection cannot be re-established within the FT Reconnection Timeout period, the LSR detecting this timeout SHOULD release all state preserved for the failed LDP session. If the TCP connection is subsequently re-established (for example, after a further Hello exchange to set up a new LDP session), the LSR MUST set the FT Reconnect Flag to 0 if it released the preserved state information on this timeout event.

If the TCP connection is successfully re-established within the FT Reconnection Timeout, both peers MUST re-issue LDP operations that were interrupted by (that is, un-acknowledged as a result of) the TCP connection failure. This procedure is described in section "FT Procedure After TCP Re-connection".

The Hold Timer for an FT LDP Session SHOULD be ignored while the FT Reconnection Timer is running. The hold timer SHOULD be restarted when the TCP connection is re-established.

4.4.1 FT LDP Operations During TCP Failure

When the LDP FT enhancements are in use for an LDP session, it is possible that an LSR may determine that it needs to send an LDP message to an LDP peer but that the TCP connection to that peer is currently down. These label operations affect the state of FT labels preserved for the failed TCP connection, so it is important that the state changes are passed to the LDP peer when the TCP connection is restored.

If an LSR determines that it needs to issue a new FT LDP operation to an LDP peer to which the TCP connection is currently failed, it MUST pend the operation (e.g. on a queue) and complete that operation with the LDP peer when the TCP connection is restored, unless the label operation is overridden by a subsequent additional operation during the TCP connection failure (see section "FT Procedure After TCP Re-connection")

In ordered operation, received FT LDP operations that cannot be correctly forwarded because of a TCP connection failure MAY be processed immediately (provided sufficient state is kept to forward the label operation) or pended for processing when the onward TCP connection is restored and the operation can be correctly forwarded upstream or downstream. Operations on existing FT labels SHOULD NOT be failed during TCP session failure.

It is RECOMMENDED that Label Request operations for new FT labels are not pended awaiting the re-establishment of TCP connection that is awaiting recovery at the time the LSR determines that it needs to issue the Label Request message. Instead, such Label Request operations SHOULD be failed and, if necessary, a notification message containing the "No LDP Connection" status code sent upstream.

Label Requests for new non-FT labels MUST be rejected during TCP connection failure, as specified in [2] and [4].

4.5 FT Procedure After TCP Re-connection

The FT operation handshaking described above means that all state changes for FT labels and Address messages are confirmed or reproducible at each LSR.

If the TCP connection between LDP peers fails but is re-connected within the FT Reconnection Timeout, both LDP peers on the connection MUST complete any label operations for FT labels that were interrupted by the failure and re-connection of the TCP connection. Label operation are completed using the procedure described below.

4.5.1 Re-Issuing FT Messages

On restoration of the TCP connection between LDP peers, any FT LDP messages that were lost because of the TCP connection failure are re-issued. The LDP peer that receives a re-issued message processes the message as if received for the first time.

"Net-zero" combinations of messages need not be re-issued after re-establishment of the TCP connection between LDP peers. This leads to the following rules for re-issuing messages that are not ACKed by the LDP peer on the LDP Initialization message exchange after re-connection of the TCP session.

- A Label Request message MUST be re-issued unless a Label Abort would be re-issued for the same Label Request or the Label Request or if the requested label is no longer required.

- A Label Mapping message MUST be re-issued unless a Label Withdraw message would be re-issued for the same FT label.

- All other messages on the LDP session that carried the FT Protection TLV MUST be re-issued if an acknowledgement had not previously been received.

Any FT label operations that were pended (see section "FT Label Operations During TCP Failure") during the TCP connection failure MUST also be issued on re-establishment of the LDP session, except where they form part of a "net-zero" combination of messages according to the above rules.

The determination of "net-zero" FT label operations according to the above rules MAY be performed on pended messages prior to the re-establishment of the TCP connection in order to optimize the use of queue resources. Messages that were sent to the LDP peer before the TCP connection failure, or pended messages that are paired with them, MUST NOT be subject to such optimization until an FT ACK TLV is received from the LDP peer. This ACK allows the LSR to identify which messages were received by the LDP peer prior to the TCP connection failure.

4.5.2 Interaction with CR-LDP LSP Modification

Re-issuing LDP messages for FT operation is orthogonal to the use of duplicate messages marked with the Modify ActFlg, as specified in [5]. Each time an LSR uses the modification procedure for an FT LSP to issue a new Label Request message, the FT label operation procedures MUST be separately applied to the new Label Request message.

5. Changes to Existing Messages

5.1 LDP Initialization Message

The LDP FT enhancements add the following optional parameters to a LDP Initialization message

Optional Parameter	Length	Value
FT Session TLV	4	See below
FT ACK TLV	4	See below

The encoding for these TLVs is found in Section "New Fields and Values".

FT Session
If present, specifies the FT behavior of the LDP session.

FT ACK TLV
If present, specifies the last FT message that the sending LDP peer was able to secure prior to the failure of the previous instantiation of the LDP session. This TLV is only present if the FT Reconnect flag is set in the FT Session TLV, in which case this TLV MUST be present.

5.2 LDP Keepalive Messages

The LDP FT enhancements add the following optional parameter to a LDP Keepalive message

Optional Parameter	Length	Value
FT ACK TLV	4	See below

The encoding for FT ACK TLV is found in Section "FT ACK TLV".

FT ACK TLV
If present, specifies the most recent FT message that the sending LDP peer has been able to secure.

5.3 All Other LDP Session Messages

The LDP FT enhancements add the following optional parameters to all other message types that flow on an LDP session after the LDP Initialization message

Optional Parameter	Length	Value
FT Protection TLV	4	See below
FT ACK TLV	4	See below

The encoding for these TLVs is found in the section "New Fields and Values".

FT Protection
If present, specifies FT Sequence Number for the LDP message.

FT ACK
If present, identifies the most recent FT LDP message ACKed by the sending LDP peer.

6. New Fields and Values

6.1 Status Codes

The following new status codes are defined to indicate various conditions specific to the LDP FT enhancements. These status codes are carried in the Status TLV of a Notification message.

The "E" column is the required setting of the Status Code E-bit; the "Status Data" column is the value of the 30-bit Status Data field in the Status Code TLV.

Note that the setting of the Status Code F-bit is at the discretion of the LSR originating the Status TLV. However, it is RECOMMENDED that the F-bit is not set on Notification messages containing status codes except "No LDP Session" because the duplication of messages SHOULD be restricted to being a per-hop behavior.

Status Code	E	Status Data
No LDP Session	0	0x000000xx
Zero FT seqnum	1	0x000000xx
Unexpected TLV / Session Not FT	1	0x000000xx
Unexpected TLV / Label Not FT	1	0x000000xx
Missing FT Protection TLV	1	0x000000xx

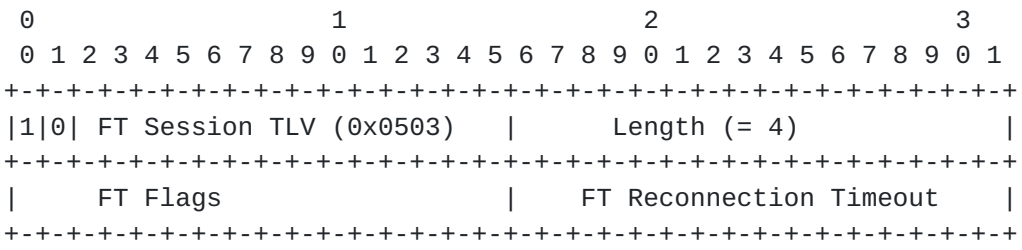
FT ACK sequence error	1	0x000000xx
Temporary Shutdown	0	0x000000xx
FT Seq Numbers Exhausted	1	0x000000xx

The Temporary Shutdown status code SHOULD be used in place of the Shutdown status code (which has the E-bit set) if the LSR that is shutting down wishes to inform its LDP peer that it expects to be able to preserve FT label state and to return to service before the FT Reconnection Timer expires.

6.2 FT Session TLV

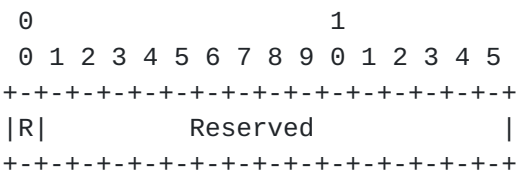
LDP peers can negotiate whether the LDP session between them supports FT extensions by using a new OPTIONAL parameter, the FT Session TLV, on LDP Initialization Messages.

The FT Session TLV is encoded as follows.



FT Flags

FT Flags: A 16 bit field that indicates various attributes the FT support on this LDP session. This fields is formatted as follows:



R: FT Reconnect Flag.
Set to 1 if the sending LSR has preserved state and resources for all FT-labels since the previous LDP session between the same LDP peers, and set to 0 otherwise. See the section "FT LDP Session Reconnection" for details of how this flag is used.

If the FT Reconnect Flag is set, the sending LSR must include an FT ACK TLV on the LDP Initialization message.

All other bits in this field are currently reserved and SHOULD be set to zero on transmission and ignored on receipt.

FT Reconnection Timeout
The period of time the sending LSR will preserve state and resources for FT labels exchanged on the previous instantiation of an FT LDP session that has currently failed. The timeout is encoded as a 16-bit unsigned integer number of seconds.

The value of 0 for this field is reserved and MUST NOT be used.

See the section "LDP Session Failure" for details of how this field is used.

6.3 FT Protection TLV

LDP peers use the FT Protection TLV to indicate that an LDP message contains an FT label operation.

The FT Protection TLV MUST NOT be used in messages flowing on an LDP session that does not support the LDP FT enhancements.

The FT Protection TLV MAY be carried on an LDP message transported on the LDP session after the initial exchange of LDP Initialization messages. In particular, this TLV MAY optionally be present on the following messages:

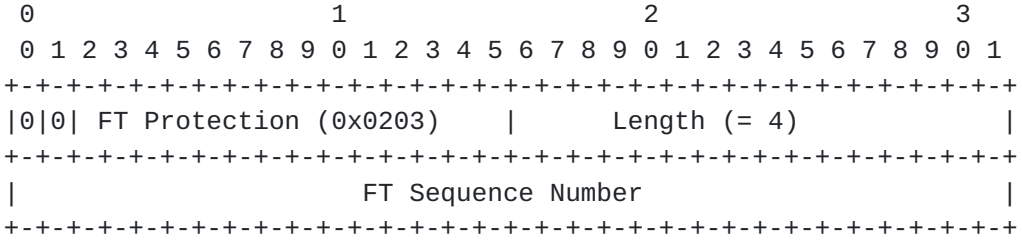
- Label Request Messages in downstream on-demand distribution mode
- Label Mapping messages in downstream unsolicited mode.

If a label is to be an FT label, then the Protection TLV MUST be present:
- on the Label Request message in DoD mode
- on the Label Mapping message in DU mode
- on all subsequent messages concerning this label.

Here 'subsequent messages concerning this label' means any message whose Label TLV specifies this label or whose Label Request Message ID TLV specifies the initial Label Request message.

If a label is not to be an FT label, then the Protection TLV MUST NOT be present on any of these messages.

The FT Protection TLV is encoded as follows.



FT Sequence Number
The sequence number for this FT label operation. The sequence number is encoded as a 32-bit unsigned integer. The initial value for this field on a new LDP session is x00000001 and is incremented by one for each FT LDP message issued by the sending LSR on this LDP session. This field may wrap from xFFFFFFFF to x00000000.

This field MUST be reset to x00000001 if either LDP peer does not set the FT Reconnect Flag on re-establishment of the TCP connection.

See the section "FT Operation Acks" for details of how this field is used.

If an LSR receives an FT Protection TLV on a session that does not support the FT LFP enhancements, it SHOULD send a Notification message to its LDP peer containing the "Unexpected TLV, Session Not FT" status code.

If an LSR receives an FT Protection TLV on an operation affecting a label that it believes is a non-FT label, it SHOULD send a Notification message to its LDP peer containing the "Unexpected TLV, Label Not FT" status code.

If an LSR receives a message without the FT Protection TLV affecting a label that it believes is an FT label, it SHOULD send a Notification message to its LDP peer containing the "Missing FT Protection TLV" status code.

If an LSR receives an FT Protection TLV containing a zero FT Sequence Number, it SHOULD send a Notification message to its LDP peer containing the "Zero FT Seqnum" status code.

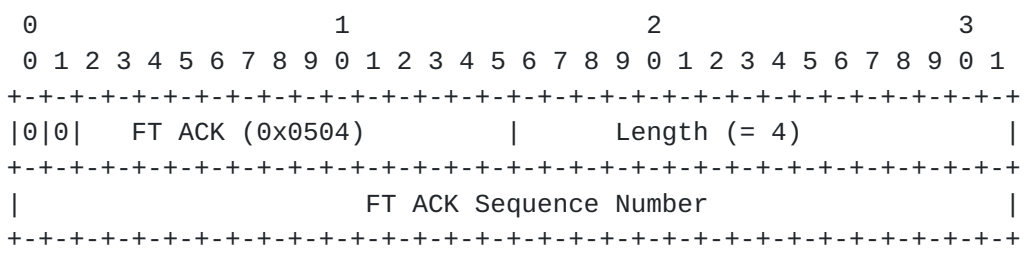
6.4 FT ACK TLV

LDP peers use the FT ACK TLV to acknowledge FT label operations.

The FT ACK TLV MUST NOT be used in messages flowing on an LDP session that does not support the LDP FT enhancements.

The FT ACK TLV MAY be present on any LDP message exchanged on an LDP session after the initial LDP Initialization messages. It is RECOMMENDED that the FT ACK TLV is included on all FT Keepalive messages in order to ensure that the LDP peers do not build up a large backlog of unacknowledged state information.

The FT ACK TLV is encoded as follows.



FT ACK Sequence Number
The sequence number for this most recent FT label message that the sending LDP peer has received from the receiving LDP peer and secured against failure of the LDP session. It is not necessary for the sending peer to have fully processed the message before ACKing it. For example, an LSR MAY ACK a Label Request message as soon as it has securely recorded the message, without waiting until it can send the Label Mapping message in response.

ACKs are cumulative. Receipt of an LDP message containing an FT ACK TLV with an FT ACK Sequence Number of 12 is treated as the acknowledgement of all messages from 1 to 12 inclusive (assuming the LDP session started with a sequence number of 1).

This field MUST be set to 0 if the LSR sending the FT ACK TLV has not received any FT label operations on this LDP session. This would apply to LDP sessions to new LDP peers or after an LSR determines that it must drop all state for a failed TCP connection.

See the section "FT Operation Acks" for details of how this field is used.

If an LSR receives a message affecting a label that it believes is an FT label and that message does not contain the FT Protection TLV, it SHOULD send a Notification message to its LDP peer containing the "Missing FT Protection TLV" status code.

If an LSR receives an FT ACK TLV that contains an FT ACK Sequence Number that is less than the previously received FT ACK Sequence Number (remembering to take account of wrapping), it SHOULD send a Notification message to its LDP peer containing the "FT ACK Sequence Error" status code.

7. Example Use

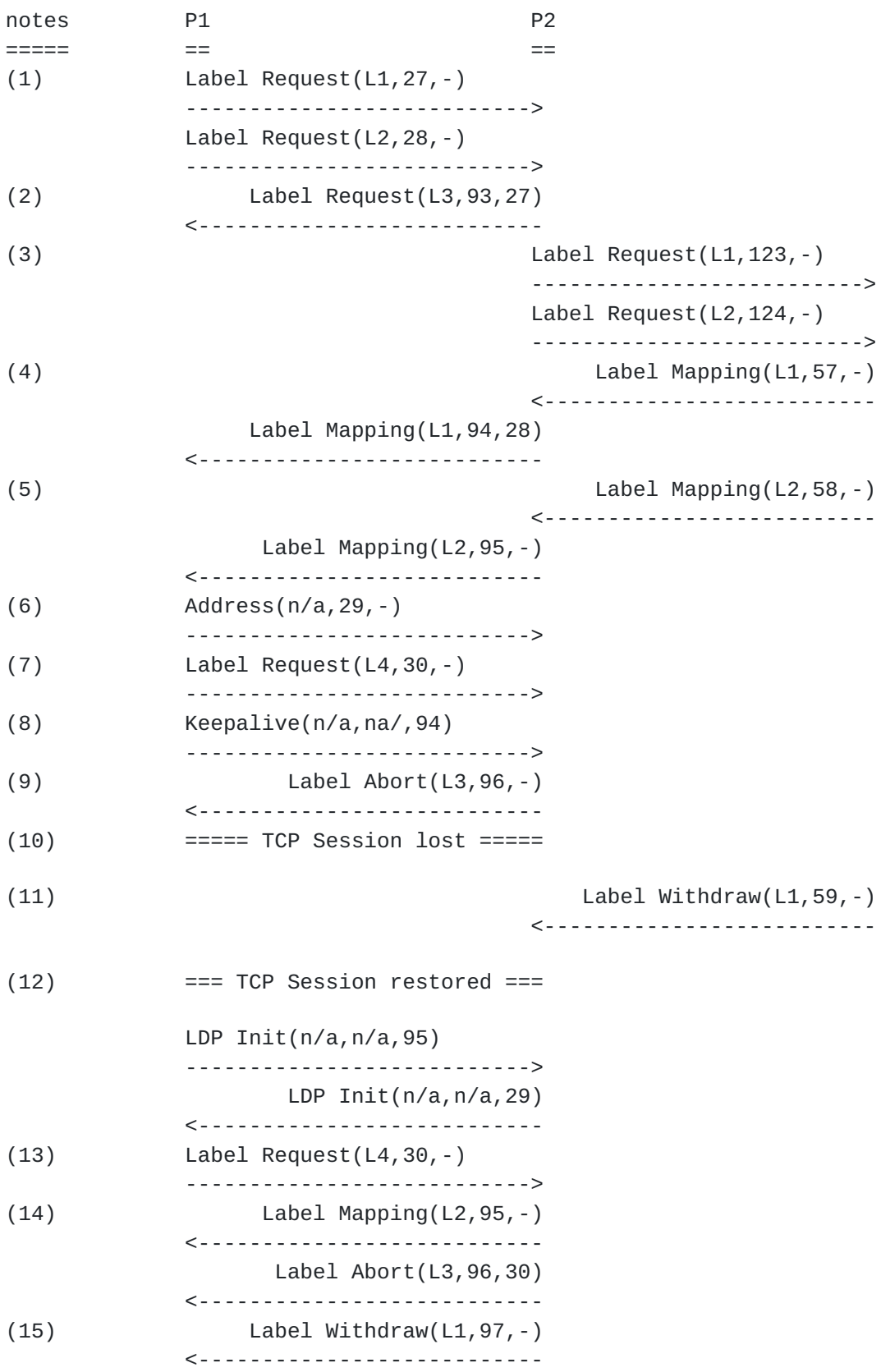
Consider two LDP peers, P1 and P2, implementing LDP over a TCP connection that connects them, and the message flow shown below.

The parameters shown on each message shown below are as follows:

message (label, senders FT sequence #, FT ACK #)

A "-" for FT ACK # means that the FT ACK TLV is not included on that message. "n/a" means that the parameter in question is not applicable to that type of message.

In the diagram below, time flows from top to bottom. The relative position of each message shows when it is transmitted. See the notes for a description of when each message is received, secured for FT or processed.


notes
P1
P2
(1)
Label Request(L1,27,-)
Label Request(L2,28,-)
(2)
Label Request(L3,93,27)
(3)
Label Request(L1,123,-)
Label Request(L2,124,-)
(4)
Label Mapping(L1,57,-)
Label Mapping(L1,94,28)
(5)
Label Mapping(L2,58,-)
Label Mapping(L2,95,-)
(6)
Address(n/a,29,-)
(7)
Label Request(L4,30,-)
(8)
Keepalive(n/a,na/,94)
(9)
Label Abort(L3,96,-)
(10)
===== TCP Session lost =====
(11)
Label Withdraw(L1,59,-)
(12)
=== TCP Session restored ===
LDP Init(n/a,n/a,95)
LDP Init(n/a,n/a,29)
(13)
Label Request(L4,30,-)
(14)
Label Mapping(L2,95,-)
Label Abort(L3,96,30)
(15)
Label Withdraw(L1,97,-)

Notes:
======

(1) Assume that the LDP session has already been initialized.
P1 issues 2 new Label Requests using the next sequence numbers.

(2) P2 issues a third Label request to P1. At the time of sending this request, P2 has secured the receipt of the label request for L1 from P1, so it includes an ACK for that message.

(3) P2 Processes the Label Requests for L1 and L2 and forwards them downstream. Details of downstream processing are not shown in the diagram above.

(4) P2 receives a Label Mapping from downstream for L1, which it forwards to P1. It includes an ACK to the Label Request for L2, as that message has now been secured and processed.

(5) P2 receives the Label Mapping for L2, which it forwards to P1. This time it does not include an ACK as it has not received any further messages from P1.

(6) Meanwhile, P1 sends a new Address Message to P2 .

(7) P1 also sends a fourth Label Request to P2

(8) P1 sends a Keepalive message to P2, on which it includes an ACK for the Label Mapping for L1, which is the latest message P1 has received and secured at the time the Keepalive is sent.

(9) P2 issues a Label Abort for L3.

(10) At this point, the TCP session goes down.

(11) While the TCP session is down, P2 receives a Label Withdraw Message for L1, which it queues.

(12) The TCP session is reconnected and P1 and P2 exchange LDP Initialization messages on the recovered session, which include ACKS for the last message each peer received and secured prior to the failure.

(13) From the LDP Init exchange, P1 determines that it needs to re-issue the Label request for L4.

(14) Similarly, P2 determines that it needs to re-issue the Label Mapping for L2 and the Label Abort.

(15) P2 issues the queued Label Withdraw to P1.

8. Security Considerations

The LDP FT enhancements inherit similar security considerations to those discussed in [2] and [4].

The LDP FT enhancements allow the re-establishment of a TCP connection between LDP peers without a full re-exchange of the attributes of established labels, which renders LSRs that implement the extensions specified in this draft vulnerable to additional denial-of-service attacks as follows:

- An intruder may impersonate an LDP peer in order to force a failure and reconnection of the TCP connection, but where the intruder does not set the FT Reconnect Flag on re-connection. This forces all FT labels to be released.

- Similarly, an intruder could set the FT Reconnect Flag on re-establishment of the TCP session without preserving the state and resources for FT labels.

- An intruder could intercept the traffic between LDP peers and override the setting of the FT Label Flag to be set to 0 for all labels.

All of these attacks may be countered by use of an authentication scheme between LDP peers, such as the MD5-based scheme outlined in [4].

Alternative authentication schemes for LDP peers are outside the scope of this draft, but could be deployed to provide enhanced security to implementations of LDP, CR-LDP and the LDP FT enhancements.

9. Implementation Notes

9.1 FT Recovery Support on Non-FT LSRs

In order to take full advantage of the FT capabilities of LSRs in the network, it may be that an LSR that does not itself contain the ability to recover from local hardware or software faults still needs to support the LDP FT enhancements described in this draft.

Consider an LSR, P1, that is an LDP peer of a fully Fault Tolerant LSR, P2. If P2 experiences a fault in the hardware or software that serves an LDP session between P1 and P2, it may fail the TCP connection between the peers. When the connection is recovered, the LSPs/labels between P1 and P2 can only be recovered if both LSRs were

applying the FT recovery procedures to the LDP session.

9.2 ACK generation logic

FT ACKs SHOULD be returned to the sending LSR as soon as is practicable in order to avoid building up a large quantity of unacknowledged state changes at the LSR. However, immediate one-for-one acknowledgements would waste bandwidth unnecessarily.

A possible implementation strategy for sending ACKs to FT LDP messages is as follows:

- An LSR secures received messages in order and tracks the sequence number of the most recently secured message, Sr.
- On each LDP KeepAlive that the LSR sends, it attaches an FT ACK TLV listing Sr
- Optionally, the LSR may attach an FT ACK TLV to any other LDP message sent between Keepalive messages if, for example, Sr has increased by more than a threshold value since the last ACK sent.

This implementation combines the bandwidth benefits of accumulating ACKs while still providing timely ACKs.

10. Acknowledgments

The work in this draft is based on the LDP and CR-LDP ideas expressed by the authors of [2] and [4].

The ACK scheme used in this draft was inspired by the proposal by David Ward and John Scudder for restarting BGP sessions [9].

The authors would also like to acknowledge the careful review and comments of Nick Weeds, Piers Finlayson, Tim Harrison and Duncan Archer at Data Connection Ltd, Peter Ashwood-Smith of Nortel and Bon Thomas of Cisco.

11. Intellectual Property Consideration

12. Full Copyright Statement

13. IANA Considerations

This draft requires the use of a number of new TLVs and status codes from the number spaces within the LDP protocol. This section explains the logic used by the authors to choose the most appropriate number space for each new entity, and is intended to assist in the determination of any final values assigned by IANA or the MPLS WG in the event that the MPLS WG chooses to advance this draft on the standards track.

This section will be removed when the TLV and status code values have been agreed with IANA.

13.1 FT Session TLV

The FT Session TLV carries attributes that affect the entire LDP session between LDP peers. It is suggested that the type for this TLV should be chosen from the 0x05xx range for TLVs that is used in [4] by other TLVs carrying session-wide attributes. At the time of this writing, the next available number in this range is 0x0503.

13.2 FT Protection TLV

The FT Protection TLV carries attributes that affect a single label exchanged between LDP peers. It is suggested that the type for this TLV should be chosen from the 0x02xx range for TLVs that is used in [4] by other TLVs carrying label attributes. At the time of this writing, the next available number in this range is 0x0203.

Consideration was given to using the message number field instead of a new FT Sequence Number field. However, the authors felt this placed unacceptable implementation constraints on the use of message number (e.g. it could no longer be used to reference a control block).

13.3 FT ACK TLV

The FT Protection TLV may ACK many label operations at once if cumulative ACKS are used. It is suggested that the type for this TLV should be chosen from the 0x05xx range for TLVs that is used in [4] by other TLVs carrying session-wide attributes. At the time of this writing, the next available number in this range is 0x0504.

Consideration was given to carrying the FT ACK Number in the FT Protection TLV, but the authors felt this would be inappropriate as many implementations may wish to carry the ACKs on Keepalive messages.

13.4 Status Codes

The authors' current understanding is that MPLS status codes are not sub-divided into specific ranges for different types of error. Hence, the numeric status code values assigned for this draft should simply be the next available values at the time of writing and may be substituted for other numeric values.

See section "Status Codes" for details of the status codes defined in this draft.

14. Authors' Addresses

Adrian Farrel (editor)
Data Connection Ltd.
Windsor House
Pepper Street
Chester
Cheshire
CH1 1DF
UK
Phone: +44-(0)-1244-313440
Fax: +44-(0)-1244-312422
Email: af@dataconnection.com

Paul Brittain
Data Connection Ltd.
Windsor House
Pepper Street
Chester
Cheshire
CH1 1DF
UK
Phone: +44-(0)-1244-313440
Fax: +44-(0)-1244-312422
Email: pjb@dataconnection.com

Philip Matthews
Nortel Networks Corp.
P.O. Box 3511 Station C,
Ottawa, ON K1Y 4H7
Canada
Phone: +1 613-768-3262
philipma@nortelnetworks.com

Eric Gray
Zaffire, Inc.
2630 Orchard Parkway,
San Jose, CA - 95134-2020
Phone: (408) 894-7362
egray@zaffire.com

15. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 Jamoussi, B., et. al., Constraint-Based LSP Setup using LDP, draft-ietf-mpls-cr-ldp-04.txt, July 2000, (work in progress).

3 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

4 Andersson, L., et. al., LDP Specification, draft-ietf-mpls-ldp-11.txt, August 2000 (work in progress).

5 Ash, G., et al., LSP Modification Using CR-LDP, draft-ietf-mpls-crlsp-modify-01.txt, February 2000 (work in progress).

6 Braden, R., et al., Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification, RFC 2205, September 1997.

7 Berger, L., et al., RSVP Refresh Reduction Extensions, draft-ietf-rsvp-refresh-reduct-05.txt, June 2000 (work in progress).

8 Swallow, G., et al,. Extensions to RSVP for LSP Tunnels, draft-

ietf-mpls-rsvp-lsp-tunnel-07.txt, August 2000 (work in progress).

9 Ward, D, et al., BGP Notification Cease: I'll Be Back, draft-ward-bgp4-ibb-00.txt, June 1999 (work in progress)

10 Stewart, R, et al., Simple Control Transmission Protocol, draft-ietf-sigtran-sctp-07.txt, March 2000 (work in progress)